

Segmentace obrazu s využitím spektrálního shlukování

Image Segmentation by Making Use of Spectral Clustering

Zadání bakalářské práce

Student:

Marek Pecha

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

1103R031 Výpočetní matematika

Téma:

Segmentace obrazu s využitím spektrálního shlukování
Image Segmentation by Making Use of Spectral Clustering

Zásady pro vypracování:

Cílem bakalářské práce je ověřit možnosti segmentace obrazu s využitím spektrální dekompozice Laplaceovy matice sítě, kterou obraz vytváří, s následným shlukováním. V diplomové práci proveďte:

- 1) Seznamte se s problematikou spektrálního shlukování (vhodným materiálem je např.: Ulrike von Luxburg, A Tutorial on Spectral Clustering).
- 2) Metodu spektrálního shlukování implementujte a experimentálně ověřte pro segmentaci obrazu. Pozornost věnujte efektivnímu postupu výpočtu vlastních čísel a vektorů Laplaceovy matice (lze použít např. knihovnu ARPACK), jakož i výběru vhodné shlukovací metody.
- 3) Metodu implementovanou dle předchozího odstavce rozšířte na tzv. difúzní spektrální shlukování (Gaura, Sojka: Diffusion Spectral Clustering for Image Segmentation).
- 4) Chování metod otestujte a shrňte dosažené výsledky.

Programátorské práce proveďte v C/C++.

Seznam doporučené odborné literatury:

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Dr. Ing. Eduard Sojka**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2014

.....

Rád bych poděkoval *doc. Dr. Eduardu Sojkovi* za vedení a poskytování cenných rad. Poděkování patří i celé mé rodině za podporu.

Abstrakt

Cílem bakalářské práce je ověřit možnosti segmentace obrazu s využitím spektrální dekompozice Laplaceovy matice sítě, kterou obraz vytváří, s následným shlukováním. V diplomové práci proveďte:

1. Seznamte se s problematikou spektrálního shlukování (vhodným materiálem je např.: Ulrike von Luxburg, A Tutorial on Spectral Clustering).
2. Metodu spektrálního shlukování implementujte a experimentálně ověřte pro segmentaci obrazu. Pozornost věnujte efektivnímu postupu výpočtu vlastních čísel a vektorů Laplaceovy matice (lze použít např. knihovnu ARPACK), jakož i výběru vhodné shlukovací metody.
3. Metodu implementovanou dle předchozího odstavce rozšiřte na tzv. difúzní spektrální shlukování (Gaura, Sojka: Diffusion Spectral Clustering for Image Segmentation).
4. Chování metod otestujte a shrňte dosažené výsledky.

Programátorské práce proveďte v C/C++.

Klíčová slova: Segmentace obrazu, spektrální shlukování, difúzní spektrální shlukování.

Abstract

The goal of this bachelor work is to verify the possibilities of image segmentation by making use of the Laplacian matrix of the graph (grid) that is created by image, which is followed by subsequent clustering. In the bachelor work, do the following:

1. Get acquainted with the technique of spectral clustering (e.g. from the paper "A tutorial on Spectral Clustering" by Ulrike von Luxburg).
2. Verify the usefulness of spectral clustering in the area of image segmentation. Pay attention to the selection of efficient method of computing eigenvalues and eigenvectors (e.g. the ARPACK library can be used). Also, pay attention to the selection of the clustering method.

3. Improve the method from the previous paragraph into the method of diffusion spectral clustering (Gaura, Sojka: Diffusion Spectral Clustering for Image Segmentation).
4. Both the methods should be tested. The results should be summarised.

The software should be created in C/C++.

Keywords: Image segmentation, spectral clustering, diffusion spectral clustering

Obsah

1	Úvod	4
2	Segmentace obrazu, shlukovací proces a jeho vlastnosti	5
2.1	The Impossibility Theorem For Clustering	6
2.1.1	Axióm 1 <i>scale-invariance</i>	7
2.1.2	Axióm 2 <i>richness</i>	7
2.1.3	Axióm 3 <i>consistency</i>	7
2.1.4	Kleinsbergova věta o nemožnosti	8
2.2	Mumfordův and Shahův funkcionál	8
3	Spektrální shlukování jako metoda pro segmentaci obrazu	15
3.1	Reprezentace obrazu	16
3.2	Grafová notace	16
3.3	Sestavení podobnostního grafu	17
3.4	Laplaceovy matice a jejich vlastnosti	20
3.4.1	Nenormovaná Laplaceova matice	21
3.4.2	Interpretace vlastních vektorů nenormované Laplaceovy matice	24
3.4.3	Normovaná Laplaceova matice, její vlastnosti a interpretace vlastních vektorů	26
3.5	Spektrální shlukování jako strategie pro segmentaci obrazu	28
3.5.1	Algoritmus nenormovaného spektrálního shlukování pro segmentaci obrazu	29
3.5.2	Algoritmus normovaného spektrálního shlukování pro segmentaci obrazu	30
3.5.3	Výpočet vlastních čísel a vektorů	31
3.5.4	Algoritmus k-means a jeho optimalizace	31

3.6	Dosažené výsledky	34
3.7	Zhodnocení dosažených výsledků	43
4	Difúzní spektrální shlukování jako strategie pro segmentaci obrazu	44
4.1	Dosažené výsledky	45
4.2	Zhodnocení testů	49
5	Praktická část	50
6	Závěr	51
7	Reference	52
	Přílohy	53
A	Příloha	53

Seznam obrázků

1	Lola bunny and Pup 1	13
2	Příklady optimálních řešení segmentační úlohy	13
3	Příklady výsledků segmentace s nevhodně nastavenými parametry	14
4	Lokálního okolí pixelu.	17
5	<i>Gaussova podobnostní funkce</i> pro různé hodnoty parametru σ	19
6	Podobnostní graf G_s jako soustava kuliček a pružin	24
7	Originál vstupu	37
8	Originál vstupu	38
9	Originál vstupu	38
10	8068.jpg	39
11	24004.jpg	40
12	27059.jpg	42
13	scgr.jpg	45
14	Originální obrázky	47

1 Úvod

Tato bakalářská práce se zabývá možnostmi spektrálního shlukování a difúzního spektrálního shlukování jako metody pro segmentaci obrazu. V první kapitole je obecně popsána problematika segmentace obrazu. Druhá kapitola je věnována spektrálnímu shlukování, jejímu popisu a prezentacím dosažených výsledků. Třetí kapitola je věnována difúznímu spektrálnímu shlukování, jejímu popisu a prezentacím dosažených výsledků. Ve čtvrté kapitole je stručně představen software, který jsem vytvořil pro potřeby testování. Dosažené výsledky jsou shrnuty v páté kapitole.

2 Segmentace obrazu, shlukovací proces a jeho vlastnosti

*I stand at the window and see a house, trees, sky. Theoretically I might say
there were 327 brightnesses and nuances of colour.*

Do I have "327"?

No. I have sky, house, and trees.

Max Wertheimer

Segmentaci obrazu chápeme jako strategii, pomocí které rozdělujeme vstupní obraz na regiony. Regionem jako takovým rozumíme nějakou část obrazu, v níž mají jednotlivé pixely podobné vlastnosti. Nejčastěji se jedná o významný objekt v popředí zachycené scény, obecně však region představuje jakoukoli oblast na obraze, ke které můžeme přiřadit smysluplný význam. I když by se teď mohlo zdát, že za výsledek můžeme považovat libovolný výstup, kterému explicitně určíme význam, opak je pravdou. Samotné segmentaci totiž zpravidla předchází co nejpřesnější specifikace úlohy, jednoduše řečeno, musíme určit, co na obraze hledáme a co budeme považovat za smysluplné řešení. Z tohoto úhlu pohledu můžeme segmentaci obrazu chápat taky jako proces *učení s učitelem*, kde učitel, člověk nikoliv *algoritmus*, koriguje jednotlivé kroky (*sub-procedurey*) a kontroluje (přijímá nebo zamítá) jejich výstupy.

Zabývat se vývojem plně automatizované, univerzální *strategie* nemá pro segmentaci obrazu smysl, neboť naše lidské vnímání světa resp. objektů v našem okolí je natolik subjektivní a rozmanité, že nelze nalézt a matematicky vyjádřit funkci, která by popisovala všechny možné způsoby rozdělení vstupního obrazu na smysluplné regiony tzn. všechny možné pojetí informací o objektech ze všech možných obrazových scén, v případě počítačového zpracování ze všech možných digitálních obrazových materiálů.

V roce 2002 profesor *Jon Kleinberg* formuloval výše uvedený problém o nemožnosti nalezení univerzální segmentační strategie ve větě známé jako *The Impossibility theorem*, volně přeloženo jako *Teorém o nemožnosti* či *Věta o nemožnosti* [2]. Hlavní myšlenka *Teorému o nemožnosti* spočívá v tvrzení, že nemůže existovat žádná segmentační metoda, která by zároveň splňovala následující tři vlastnosti *stálost* (*consistency*), *bohatost* (*richness*) a aby byla *invariantní vůči změně měřítka* *distanční funkce* (*scale-invariance*).

Než přejdu k zavedení *Kleinbergovy věty*, představím ještě jednu techniku z oblasti shlukové analýzy, která je základem segmentačních metod, jejichž vhodnost pro segmentaci obrazu zkoumám ve své práci. Touto technikou je shlukování (*clustering*).

Definice 2.1 Shlukování

Mějme množinu rozumných vstupních dat

$$S = \{x_1, x_2 \dots x_n\} \wedge |S| \geq 2. \quad (1)$$

Shlukováním rozumíme proces, který na množině S detekuje podobnosti mezi jednotlivými prvky a z této množiny extrahuje množinu podmnožin $\Omega = \{C_1, \dots, C_k, 1 \leq k \leq n\}$ takovou, že prvky uvnitř množiny C_i jsou si navzájem maximálně podobné a zároveň nejsou příliš podobné s prvky mimo tuto množinu tj. s prvky v množinách $\Omega \setminus C_i$. Množinu C_i nazveme komponentou rozkladu množiny S a platí:

$$C_i \cap C_j = \emptyset \wedge C_1 \cup C_2 \dots C_k = S \quad (2)$$

Množinu C_i budeme dále označovat jako shluk a prvky množiny C_i jako členy shluku, množinu podmnožin Ω nazveme rozkladem množiny S .¹

Pro úvahy ve zbývající části textu nebude třeba zavádět Kleinbergovu větu o nemožnosti obecně, ale vhodnější bude její představení pro metodu shlukování.

2.1 The Impossibility Theorem For Clustering

Mějme opět nějakou množinu S (viz eq. 1) rozumných vstupních dat, a na této množině definujme párovou distanční funkci $d : S \times S \rightarrow R$ takovou, že:

$$d(x_i, x_j) \begin{cases} = 0 & x_i = x_j \\ \geq 0 & \text{jinak} \end{cases} \quad (3)$$

Následující požadavek na distanční funkci d sice není nutnou podmínkou, ale pro účely segmentace obrazových dat budeme předpokládat, že distanční funkce splňuje trojúhelníkovou nerovnost:

$$d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k) \quad (4)$$

Nyní zavedme slukovací funkci f jako unární funkci, jejíž argumentem je párová distanční funkce d a výstupem (hodnotou) je Ω (rozklad množiny S) tj. $f : d \rightarrow \Omega$. Shlukovací funkci budeme dále značit jako $f(d)$, pokud nebude řečeno jinak. Dále buď $\text{range}(f)$ množina všech možných rozkladů Ω pro libovolnou distanční funkci d .

V této chvíli již máme k dispozici potřebný matematický aparát k zavedení jednotlivých axiomatických vlastností, které budeme požadovat po univerzální shlukovací funkci $f(d)$.

¹V některých odborných publikacích se můžeme setkat, že *shlukování* je zaváděno ve smyslu *segmentace*. Ve své práci však striktně rozlišuji tyto dva pojmy a shlukování chápu jako konkrétní strategii segmentace.

2.1.1 Axióm 1 *scale-invariance*

Pro libovolnou distanční funkci d a nějaké $\alpha : \alpha > 0, \alpha \in R$, platí $f(d) = f(\alpha \cdot d)$.

Jinými slovy, shlukovací funkce $f(d)$ by neměla být citlivá na změnu jednotkové míry distanční funkce d .

2.1.2 Axióm 2 *richness*

$range(f)$ se rovná množině všech možných rozkladů S .

Neboli pro libovolný rozklad Ω je možno nalézt distanční funkci d , takovou že $f(d) = \Omega$.

2.1.3 Axióm 3 *consistency*

Posledním diskutovaným axiomem, který by měla splňovat univerzální shlukovací funkce, je *stálost*. Tím, že řekneme, že shlukovací funkce $f(d)$ je stálá, máme na mysli následující vlastnost:

Pokud zmenšíme vzdálenosti mezi členy uvnitř shluku C_k a zvětšíme vzdálenosti mezi jednotlivými shluky $C_1, C_2 \dots C_n$ měly bychom dostat stejný rozklad množiny S .

Pro preciznější zadefinování předcházejícího požadavku zavedeme operaci Ω -transformace:

Mějme dvě distanční funkce d a d' . Řekneme, že d' je Ω -transformací d pokud platí:

- a) *pro všechny $x_i, x_j \in S$ náležící stejnému shluku $C_k \in \Omega$, pak $d'(x_i, x_j) \leq d(x_i, x_j)$*
- b) *pro všechny $x_i, x_j \in S$ náležící dvou různým shlukům $C_k, C_m \in \Omega$, pak $d'(x_i, x_j) \geq d(x_i, x_j)$*

Nyní můžeme **Axióm 3** vyjádřit následovně:

Bud' d a d' dvě distanční funkce. Pokud $f(d) = \Omega$ a d' je Ω -transformací d , pak $f(d') = \Omega$.

Pomocí předcházejících třech *axiomů* již není těžké zavést *Kleinsbergovu větu o nemožnosti*:

2.1.4 Kleinsbergova věta o nemožnosti

Pro libovolnou množinu $S = \{x_1, x_2 \dots x_n\} \wedge |S| \geq 2$ neexistuje žádná shlukovací funkce $f(d)$, která by zároveň byla *bohatá*, *stálá* a *invariantní vůči změně měřítka* *distanční funkce* d .

Dále již není třeba se detailněji zabývat jednotlivými axiomy a zmíněnou *Větou o nemožnosti*, neboť máme k dispozici matematicky korektní definici, která shrnuje hypotézu o neexistenci univerzální shlukovací funkce, takže i strategie, kterou jsme zavedli v úvodu této kapitoly.

I když už víme, že nalézt univerzální shlukovací funkci je nemožné, má smysl zkoumat jiné úlohy, a to, zabývat se možnostmi a kvalitou výsledků konkrétních shlukovacích funkcí metod. Cílem mojí práce je prozkoumat využitelnost *spektrálního shlukování* a *difúzního spektrálního shlukování* v aplikaci na segmentaci obrazu. Pro shlukovací proces v těchto segmentačních strategiích využívám metody, založené na iterativním procházení prostoru bodů a jejich přiřazování k nejbližším *centroidům*. Konkrétně pro tyto typy metod neplatí **axióm 3**, důkaz lze nalézt v [2, str. 7].

Detailněji představím *spektrální shlukování* v kapitole 3. Kapitola 4 je věnována rozšířením spektrálního shlukování na *difúzní spektrální shlukování*. Než se však začnu zabývat jejich popisem, rád bych ukázal ještě jedno pojetí segmentace obrazu, které podle mě tvoří pomyslný myšlenkový most spojující obecné pojetí segmentace obrazu se spektrálním shlukováním, konkrétně se dá ukázat souvislost vlastních vektorů *Laplaceovy matice* s řešením spojitě *Laplaceovy rovnice*.

2.2 Mumfordův and Shahův funkcionál

V roce 1989 dva matematikové *David Mumford* a *Jayant Shah* navrhli funkcionál, který popisuje obecné optimální kritéria pro segmentaci obrazu a který je dnes využíván především v *level-set* metodách, založených na minimalizaci tohoto funkcionálu.

Samozřejmě, že tyto metody nejsou předmětem našeho zájmu. Co je však z mého pohledu daleko přínosnější je fakt, že se dá ukázat jistá souvislost mezi *Mumfordovým* a *Shahovým* pojetím segmentace obrazu a spektrálním shlukováním.

Pro účely mojí bakalářské práce je vhodnější uvažovat *Mumfordův a Shahův funkcionál*, dále jako F_{MS} , pokud nebude řečeno jinak, ve zjednodušeném tvaru, neboť podrobnější zkoumání komplikovanějších vyjádření by zbytečně rozvádělo text do témat, které by s mojí prací přímo nesouvisely.

Ve svém návrhu dizertační práce J. Gaura představil *Mumfordův a Shahův funkcionál* právě v takové podobě, jenž je pro pochopení myšlenky obecných optimálních kritérií pro segmentaci obrazu nanejvýš dostačující:

$$F_{MS} = \alpha \int_{\psi} [g(x, y) - f(x, y)]^2 dS + \beta \int_{\psi \setminus B} |\nabla f(x, y)|^2 dS + \gamma |B|, \quad (5)$$

kde ψ je oblast obrazu, $g(x, y)$ zastupuje originální obrazovou funkci, $f(x, y)$ je nová po částech hladká reprezentace obrazu, která minimalizuje funkcionál F_{MS} , a jednotlivé po částech hladké oblasti chápeme jako významné objekty na obraze. B představuje množinu hranic mezi jednotlivými oblastmi a α, β, γ jsou skaláry, které determinují chování funkcionálu a ovlivňují výsledek segmentace obrazu.

Nyní zaveďme úlohu minimalizace (5):

$$\min_{B, f(x, y)} F_{MS} = \min_{B, f(x, y)} \left(\alpha \int_{\psi} [g(x, y) - f(x, y)]^2 dS + \beta \int_{\psi \setminus B} |\nabla f(x, y)|^2 dS + \gamma |B| \right) \quad (6)$$

pomocí které sestavíme *obecná optimální kritéria* pro oblasti na obraze zastupující reálné významné objekty na zachycené scéně tj. *obecná optimální kritéria* pro množinu hranic B mezi jednotlivými oblastmi na ψ a pro novou reprezentaci $f(x, y)$ originálního obrazu $g(x, y)$:

- a) segmentovaný obraz $f(x, y)$ by měl být, co možná nejpodobnější s originálním obrazem $g(x, y)$. Tento požadavek popisuje první člen F_{MS} :

$$\int_{\psi} [g(x, y) - f(x, y)]^2 dS \quad (7)$$

- b) s výjimkou hraničních bodů objektů, tj. $\psi \setminus B$, by měla být čtvercová změna funkčních hodnot obrazové funkce segmentovaného obrazu $f(x, y)$ co nejmenší:

$$\int_{\psi \setminus B} |\nabla f(x, y)|^2 dS \quad (8)$$

toto kritérium můžeme ekvivalentně formulovat taky jako požadavek, že změny hodnot funkce $f(x, y)$ na hranici $b : b \in B$ mezi dvěma objekty mají být co největší.

c) Z posledního členu F_{MS}

$$|B| \quad (9)$$

explicitně vyplývá, že po množině hranic B mezi jednotlivými po částech hladkými oblastmi na ψ požadujeme, aby součet délek všech hranic $b \in B$ byl co možná nejkratší.

d) Různé rozumné hodnoty skalárů α, β, γ ilustrují fakt, že neexistuje pouze jediné smysluplné řešení segmentace daného obrazu, ale volba správného výsledku je subjektivní záležitostí (viz kap. 2.1).

Dá se ukázat, že odstraněním libovolného členu z funkcionálu F_{MS} může vést k tomu, že výsledné řešení nemusí být akceptovatelné, neboť nebudou splněny všechny podmínky optimality. Jelikož pro nás v tuto chvíli není podstatné nalezení optimálního řešení, ale budeme se snažit vyjádřit vztah mezi takto zavedeným obecným pojetím segmentace obrazu a spektrálním shlukováním, mohu provést následující sérii úvah:

- Předpokládejme, že máme nějaké smysluplné řešení $f(x,y)$ libovolného rozumného obrazu $g(x,y)$, které nemusí splňovat všechny podmínky optimality. Potom oblasti zastupující reálné objekty na ψ jsou determinovány množinou hranic B , které nemusí mít podle předcházejícího předpokladu nejkratší možnou délku. Funkcionál F_{MS} za těchto podmínek můžeme tedy přepsat do tvaru

$$F_{MS} \sim F'_{MS} = \alpha \int_{\psi} [g(x,y) - f(x,y)]^2 dS + \beta \int_{\psi \setminus B} |\nabla f(x,y)|^2 dS \quad (10)$$

- Jelikož $f(x,y)$ nepředstavuje neoptimálnější interpretaci objektů na ψ , potom ani nemusí být nutně splněna první podmínka optimality vyjadřující, že segmentovaný obraz by měl být co možná nejpodobnější s originálním obrazem $g(x,y)$. Proto funkcionál F'_{MS} za této podmínky můžeme napsat v následující podobě

$$F'_{MS} \sim F''_{MS} = \beta \int_{\psi \setminus B} |\nabla f(x,y)|^2 dS \quad (11)$$

Jediným kritériem, které tedy požadujeme po výsledných shlucích je, aby čtvercová změna hodnot uvnitř shluků $\psi \setminus B$ byla co nejmenší. Není těžké si rozmyslet, že tento požadavek je analogií shlukování, které jsme definovali na diskretní množině vstupních dat [viz def. 2.1].

Nyní vyřešme minimalizaci funkcionálu (6):

$$\min_{f(x,y)} F''_{MS} = \min_{f(x,y)} \beta \int_{\psi \setminus B} |\nabla f(x,y)|^2 dS \sim \quad (12)$$

$$\sim \beta \int_{\psi \setminus B} \min_{f(x,y)} |\nabla f(x,y)|^2 dS \sim \quad (13)$$

$$\sim \beta \int_{\psi \setminus B} \min_{f(x,y)} |\nabla f(x,y)| \cdot \min_{f(x,y)} |\nabla f(x,y)| dS \sim \quad (14)$$

$$\sim \beta \int_{\psi \setminus B} \min_{f(x,y)} \nabla f(x,y) \cdot \min_{f(x,y)} \nabla f(x,y) dS \sim \quad (15)$$

$$\sim \beta \int_{\psi \setminus B} \min_{f(x,y)} (\nabla f(x,y) \cdot \nabla f(x,y)) dS \sim \quad (16)$$

$$\sim \beta \int_{\psi \setminus B} \min_{f(x,y)} (\Delta f(x,y)) dS \sim \quad (17)$$

$$\sim \beta \min_{f(x,y)} \int_{\psi \setminus B} (\Delta f(x,y)) dS \quad (18)$$

Z (18) již jednoduše určíme řešení minimalizační úlohy (12), provedme následující úvahu:

Úloha (18) bude mít minimální řešení právě tedy když, laplaceův diferenciální operátor Δ na funkci $f(x, y)$ bude roven 0 tj.

$$\beta \min_{f(x,y)} \int_{\psi \setminus B} (\Delta f(x,y)) dS \iff \beta \Delta f(x,y) = 0 \quad (19)$$

Jinými slovy, úloha (12) má minimální řešení pouze pro takové funkce $f(x, y)$, které jsou řešením *homogenní Laplaceovy rovnice* za předem daných okrajových podmínek. Zatím předpokládejme, že pro tuto úlohu máme stanoveny nulové počáteční podmínky, buď *Dirichletovy* či *Dirichletovy* a *Neumannovy*, podle dané fyzikální interpretace. Dále budeme předpokládat, že $f(x, y)$ a $k \cdot f(x, y)$, $k \in \mathbb{R}$ nám budou předávat stejnou informační hodnotu tj. $f(x, y) \sim k \cdot f(x, y)$. Potom můžeme *homogenní Laplaceovu rovnici* na pravé straně ekvivalence (19) upravit:

$$\beta \Delta f(x,y) = 0 \sim \Delta f(x,y) = 0 \quad (20)$$

Jelikož není vhodné strojově zpracovávat spojité úlohy, především díky vysokým paměťovým nárokům, musíme rovnici (20) převést do diskretní domény. Označme si $\Delta f(x, y)$ jako L , což budeme chápat jako diskretní tvar spojitého diferenciálního operátoru Δ na funkci $f(x, y)$, a x nějaké diskretní řešení (20). Pak můžeme *homogenní Laplaceovu rovnici* diskretizovat následovně:

$$L \cdot x = o \quad (21)$$

Ještě upravíme pravou stranu rovnice (21). o je nulový vektor, ten můžeme ekvivalentně vyjádřit dvěma způsoby $o = k \cdot o$, kde $k \in \mathbb{R}$, nebo případně můžeme použít druhý způsob vyjádření $o = 0 \cdot x$. Pro účely tohoto textu bude vhodnější použít druhý způsob, takže můžeme psát:

$$L \cdot x = 0 \cdot x \quad (22)$$

Jelikož 0 je konstanta označme si ji např. λ :

$$L \cdot x = \lambda \cdot x \quad (23)$$

Nyní už není těžké poznat, že (23) je vlastně diskretním vyjádřením *Laplaceovy rovnice* s nulovými počátečními podmínkami ve tvaru řešení *problému vlastních čísel a vektorů* matice L , která bývá označována jako *Laplaceova*. V dalších částech textu bude ukázáno, že pouze pro $\lambda = 0$ jsou příslušné vlastní vektory diskretním řešením *Laplaceovy rovnice* (20)¹ a právě ty budeme v klasickém přístupu ke *spektrálnímu shlukování* zkoumat a využívat pro detekci objektů na obraze. Již teď bychom si však měli být vědomi několika problémů, které vyplývají z předchozích poznatků, a k řešení segmentace obrazu pomocí *spektrálního shlukování* bychom měli přistupovat dosti obezřetně:

- kromě toho, že může existovat více řešení $f(x, y)$ představující nějakou segmentaci daného obrazu (*viz kap. 2.1*), některé z těchto řešení pro dané nastavení parametrů mohou být nesmyslné, díky tomu, že nepředstavují optimální řešení. Konkrétně nemáme zaručeno podmínku, že segmentovaný obraz $f(x, y)$ bude mít součet všech délek hranic mezi jednotlivými objekty minimální (*eq. 10*). Co je však pro segmentaci obrazu dosti kritické je fakt, že nemáme zaručeno, aby byl výsledný segmentovaný obraz co nejvíce podobný s originálním (*eq. 11*), proto můžeme očekávat libovolný výstup pro libovolné nastavení parametrů metody *spektrálního shlukování*. I když ještě nemáme korektně zavedenou metodu *spektrálního shlukování*, rád bych v **Př. 1** ukázal jaké vhodné a nesmyslné výsledky můžeme očekávat.

¹Laplaceova rovnice s nulovými počátečními podmínkami, tak jak jsme ji v tomto textu definovali, má pouze jediné řešení a to konstantní funkci. Proto k -násobnost vlastního čísla $\lambda = 0$ znamená, že *vlastní problém*(23) představoval řešení soustavy k zdiskretizovaných Laplaceových rovnic a každý jednotlivý vlastní vektor x reprezentuje jeden objekt na obraze. Výsledný segmentovaný obraz $f(x, y)$ potom získáme složením jednotlivých vlastních vektorů x , například pomocí nějakého shlukovacího algoritmu.

Př. 1 Mějme následující digitálně nakreslený obraz. Záměrně jsem jako první ukázkou zvolil kreslený obraz neboť u takovýchto obrázků není třeba řešit problém s mělkými přechody, se kterými se setkáme u zpracování reálných obrazových materiálů - fotek.



Obrázek 1: Lola bunny and Pup 1

CHOZAN, Aung. <http://chozan.deviantart.com> [online]. [cit. 14.4.2014].

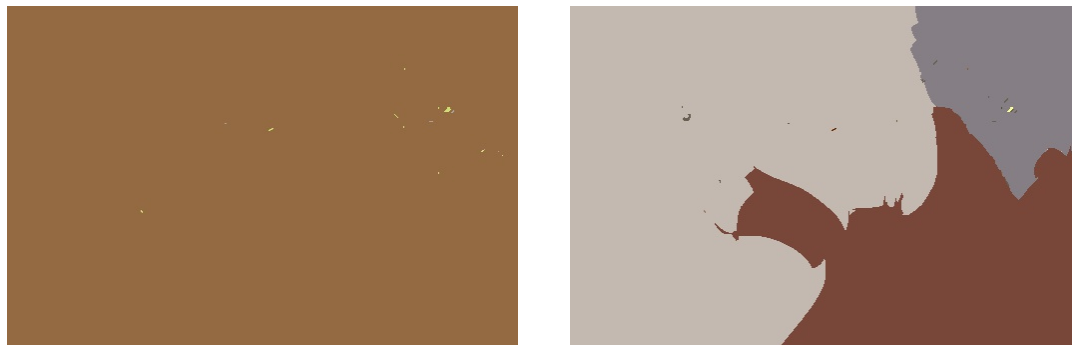
Dostupný na www: <http://fav.me/d529m5q>

Optimální řešení segmentace zadaného obrazu mohou vypadat např. takto:



Obrázek 2: Příklady optimálních řešení segmentační úlohy

Pokud však zvolíme nevhodné nastavení parametrů může metoda *spektrálního shlukování* vracet výsledky, které nám nepředají žádnou informační hodnotu o objektech na zachycené scéně:



Obrázek 3: Příklady výsledků segmentace s nevhodně nastavenými parametry

3 Spektrální shlukování jako metoda pro segmentaci obrazu

*"When you can measure what you are speaking
about and express it in numbers, you know
something about it;
but when you cannot measure it,
when you cannot express it in numbers,
your knowledge is of the meager and unsatisfactory kind."*
Lord Kevin

Cílem této kapitoly je zavedení a hlubší prostudování metody *spektrálního shlukování* v souvislosti s úlohou segmentace obrazu. Rád bych upozornil, že se v průběhu seznamování s naší aplikací *spektrálního shlukování* nevyhneme několika problémům, se kterými je třeba se co nejlépe vypořádat.

Jako první úskalí se mohou jevit různé interpretace jednotlivých kroků, což není až tak způsobeno metodou *spektrálního shlukování*, ale tento problém vyplývá ze samotné práce s digitálním obrazem. Ve své podstatě je *analýza obrazu* natolik obecná disciplína, že je třeba hledat vhodné analogie buď v konkrétních inženýrských úlohách nebo nahlížet na problematiku z různých matematických přístupů. Další komplikací, na kterou při zavádění metody narazíme, je, že nesmíme zapomenout zakomponovat do strojového zpracování segmentace obrazu *lidský faktor* tak, aby bylo možné měnit nastavení výchozích parametrů segmentační úlohy tj. zachovat myšlenku, že segmentace obrazu má být subjektivní záležitostí.

Tyto důvody mě vedly k nápadu, že nejrozumnější způsob, jak zamezit promíchávání interpretací a nekorektnostem při zavádění definic, je rozdělit celý proces *spektrálního shlukování* do několika podúloh a každé jednotlivé podúloze se věnovat v samostatné kapitole.

Aby bylo možné, co nejpřesněji vyjádřit vlastnosti námi zkoumaného přístupu k segmentaci obrazu, zvolil jsem pro hledání vhodných analogií dva různé přístupy. Pomocí prvního budeme nahlížet na problematiku jako na spojitý problém pomocí matematického aparátu *parciálních diferenciálních rovnic*, který nám bude sloužit spíš k hlubšímu pochopení jednotlivých souvislostí než k vlastnímu strojovému zpracování. Pro vlastní algoritmizaci metody budou vhodnější postupy převzaté z *teorie grafů*, neboť jak jsem se již zmínil v předchozí kapitole, zpracovávat spojitě problémy na současných počítačích je z praktického hlediska nevhodné.

3.1 Reprezentace obrazu

Mějme obraz jako pole pixelů $P = [p_1, p_2 \dots p_n]$, jejichž hodnoty mohou být vyjádřeny buď *skalárem*, v případě že pracujeme s obrazem v odstínech šedi, nebo tří složkovým vektorem *RGB*. Takovouto reprezentaci si můžeme představit například tak, že obraz máme uložený v lineární paměti počítače.

Cílem shlukování, jak víme z Definice 2.1, je snaha extrahovat ze vstupní množiny dat takové podmnožiny, ve kterých jsou si prvky maximálně podobné. Problémem při zpracování obrazu je, že nemáme primárně k dispozici žádné informace o podobnostech mezi pixely. V takovýchto případech je vhodné modelovat data ve formě podobnostního grafu $G_s = (V, E)$, kde každý vrchol v_i zastupuje jeden prvek ze vstupní množiny, v našem případě pixel p_i . Dva vrcholy v_i a v_j spojíme hranou e_{ij} , pokud má smysl zabývat se podobností mezi pixely p_i a p_j nebo pokud je jejich vzájemná podobnost $s_{ij} \in S$ větší než nějaký předem zvolený práh. Hodnotou s_{ij} budeme považovat za váhu w_{ij} hrany e_{ij} .

Nyní můžeme problém shlukování intuitivně přeformulovat pomocí úlohy řezu grafů: snažíme se najít takovou komponentu grafu, že váhy hran uvnitř této komponenty mají dostatečně vysoké hodnoty a hodnoty vah hran mezi jednotlivými komponentami jsou dostatečně nízké. Pro shlukování z pohledu teorie grafů požadujeme opačný požadavek než jsme požadovali po shlukování v Definici 2.1, proto bude třeba pro vyjádření míry podobnosti mezi pixely použít jinou podobnostní funkci, jelikož párová distanční funkce d nesplňuje potřebnou vlastnost.

Abychom bylo schopni formalizovat toto intuitivní zavedení, je třeba zavést příslušné matematické značení.

3.2 Grafová notace

Buď $G_s = (V, E)$ neorientovaný vážený graf s množinou vrcholů $V = \{v_1, v_2, \dots, v_n\}$ a množinou hran $E = \{e_1, e_2, \dots, e_k\}$. Předpokládejme, že váhy w_{ij} budou nabývat pouze nezáporných hodnot $w_{ij} \geq 0$. Dále buď W matice sousednosti grafu G_s taková, že:

$$W = (w_{ij})_{i,j=1,2,\dots,n} \quad (24)$$

Pokud prvek w_{ij} matice W je roven 0, znamená to, že vrcholy v_i a v_j nejsou spojené hranou. Jelikož uvažujeme neorientovaný graf G_s navíc platí $w_{ij} = w_{ji}$. Dále definujeme stupeň vrcholu $d_i = \sum_{j=1}^n w_{ij}$ a diagonální matici D s prvky d_1, d_2, \dots, d_n na hlavní diagonále:

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & d_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & d_n \end{bmatrix} \quad (25)$$

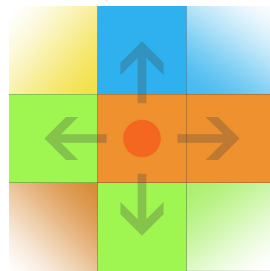
Dále zavedme podmnožinu vrcholů $A \subset V$ a její doplněk $\bar{A} = V \setminus A$. Množinu A budeme považovat za takovou komponentu grafu G_s , že hrany uvnitř této komponenty mají dostatečně vysoké hodnoty vah, a budeme ji chápat jako analogii komponenty rozkladu C_i zavedenou v Definici 2.1.

Označme si $f_{A_j} \in \mathbb{R}^n$ vektor, jehož prvek $f_{A_j i} = 1$ pokud vrchol $v_i \in A_j$, jinak je jeho hodnota rovna 0. Vektor f budeme označovat termínem *indikátorový vektor* a v další části textu bude často zmiňován v souvislosti s vlastními vektory *Laplaceových matic* příslušných nulovým vlastním číslem.

3.3 Sestavení podobnostního grafu

Prvním nápadem na vytvoření podobnostního grafu G_s může být, že na množině vrcholů vytvoříme kompletní graf tj. takový neorientovaný graf, ve kterém jsou každé dva vrcholy spojené hranou. Obecně je takový přístup vhodný pouze pro malé množiny vstupních dat. My však budeme pracovat s obrazy, které mají počet pixelů řádově 10^4 až 1.5^5 . Pokud bychom do matice W museli ukládat všech $|P|^2$ hodnot, potom velikost W pro $|P| = 10^4$ by byla $\sim 385MB$, pro druhý krajní případ $|P| = 1.5^5$ by už byla paměťová náročnost matice $\sim 85GB$, což je pro běh úlohy na obyčejném osobním počítači naprosto nevhodná reprezentace. Vhodnější je namodelovat podobnostní graf G_s pomocí lokálních ϵ -okolích dat.

V článku [1, str. 3] jsou diskutovány různé způsoby, jak sestavit podobnostní graf G_s , abychom měli zaručeno vhodné namodelování lokálních okolí dat. V našem případě bude vhodnější využít znalosti původu vstupních dat, díky které víme, že obraz je na grafickém výstupu zobrazován jako matice obrazových bodů, v níž je každý pixel determinován hodnotou a xy -souřadnicí. Za lokální okolí je tedy velice vhodné zvolit pravého, levého, horního a dolního souseda daného pixelu:



Obrázek 4: Lokálního okolí pixelu.

V technikách zpracování obrazu se často používají speciální typy grafů zvané *mesh grafy*, ve kterých má každý vrchol všechny 4 sousedy. Pro naše účely bude postačovat mřížkový graf, v něm vrcholy na topologickém okraji grafu nemají kompletní lokální 4-okolí, ale

některý soused danému vrcholu chybí, což koresponduje s vlastností obrazové matice. Libovolný obraz můžeme převést na *mřížkový graf* pomocí následujícího algoritmu:

Input: $P = \{p_1, p_2 \dots p_n\}$ jako pole pixelů; w jako šířka obrázku; h jako výška obrázku;

Output: G_s jako podobnostní graf vytvořený z množiny pixelů

for $i \leftarrow 1$ **to** $w * h$ **do**

 | Vytvoř vrchol v_i a obarvi jej příslušnou barvou pixelu p_i .

end

for $y \leftarrow 1$ **to** h **do**

for $x \leftarrow 1$ **to** w **do**

$i \leftarrow x + (w - 1) * y$;

if $x \neq W$ **then**

 | $j \leftarrow i + 1$; Vytvoř hranu e_{ij} .

else if $x \neq 1$ **then**

 | $j \leftarrow i - 1$; Vytvoř hranu e_{ij} .

else if $y \neq h$ **then**

 | $j \leftarrow i - w$; Vytvoř hranu e_{ij} .

else if $y \neq 1$ **then**

 | $j \leftarrow i + w$; Vytvoř hranu e_{ij} .

end

end

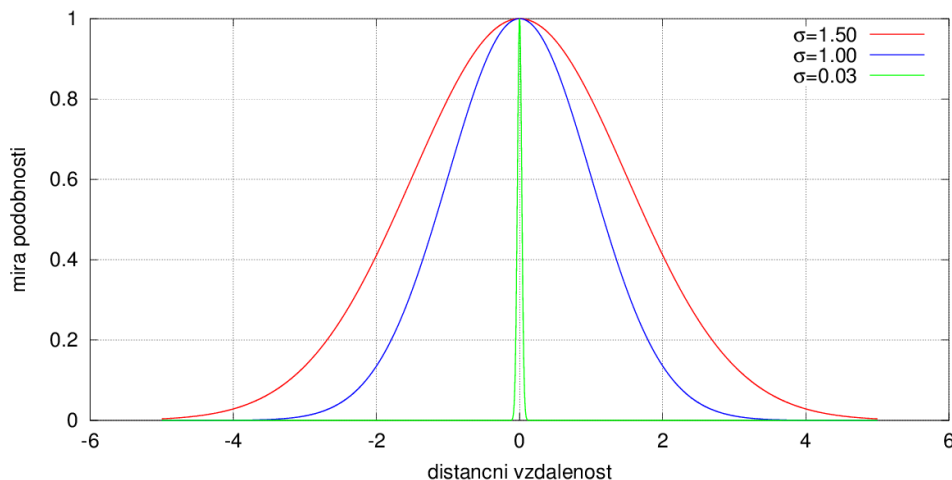
end

Algoritmus 1: Transformace vstupního obrazu na podobnostní graf G_s

Dále je třeba určit váhy w_{ij} hran e_{ij} grafu G_s jako podobnosti s_{ij} hodnot vrcholů v_i . Za hodnotu vrcholu v_i považujeme v našem případě hodnotu jasu případně barvy v pixelu p_i . Jako vhodnou funkci pro výpočet podobnosti můžeme zvolit *Gaussovu podobnostní funkci*

$$gauss_{sim}(p_i, p_j) = e^{-\frac{d(p_i, p_j)^2}{2\sigma^2}} \quad (26)$$

kde $d(p_i, p_j)$ je párová distanční funkce s *Euklidovskou metrikou* a σ je konstanta popisující rozptyl lokálního ϵ -okolí. V našem případě můžeme σ chápat jako volný parametr ovlivňující detailnost výsledné segmentace, čím bude hodnota σ nižší, tím detailnější bude výsledná segmentace. Typický průběh *Gaussovy podobnostní funkce* pro různé hodnoty rozptylu je zobrazen na obrázku 5.



Obrázek 5: Gaussova podobnostní funkce pro různé hodnoty parametru σ

Jak je z grafů zřetelné pro rostoucí vzdálenost mezi hodnotami pixelů a se snižující se hodnotou parametru σ velice rychle klesá míra podobnosti $gauss_{sim}(p_i, p_j)$ mezi pixely p_i a p_j a to tak, že pro vzdálenost $|d(p_i, p_j)| \sim 6$ a hodnotu parametru $\sigma = 1.5$ je míra podobnosti $gauss_{sim}(p_i, p_j) \sim 0$. Pokud bychom nastavili hodnotu parametru σ na hodnotu 0.03, tak už pro vzdálenost $|d(p_i, p_j)| \geq 1$ je míra podobnosti $gauss_{sim}(p_i, p_j) \sim 0$.

Uvědomme si, že v obrazu ve stupních šedi nabývá skalár reprezentující hodnotu jasu pixelu hodnot $1, 2 \dots 255$ a stejných hodnot nabývá i každá složka vektoru RGB reprezentující hodnotu barevného pixelu, proto je třeba si pečlivě rozmyslet jakou volit hodnotu rozptylu σ , abychom pomocí Gaussovy podobnostní funkce dokázali smysluplně modelovat podobnosti na lokálním okolí jednotlivých pixelů. První způsob může být volba hodnoty $\sigma \gg 1$, tím zajistíme, že *gaussián* bude klesat k 0 daleko pomaleji. Druhým způsobem, podle mě daleko vhodnějším, je hodnoty vstupního obrazu znormovat tj. udělat projekci každé barevné složky, popřípadě hodnot jasu, do intervalu $\langle 0, 1 \rangle$ a podle pravidla 3σ volit hodnotu z rozsahu $\langle 0, 0.3 \rangle$.

Způsobů jak znormovat obraz je více. Jako první přístup můžeme zvolit postup, kdy každou hodnotu pixelu obrazu ve stupních šedi resp. každou hodnotu jednotlivé složky vektoru RGB podělíme číslem 255, tím budeme mít zaručeno, že všechny hodnoty pixelů obrazu ve stupních šedi resp. všechny hodnoty jednotlivých barevných složek budou ležet v intervalu $\langle 0, 1 \rangle$.

Podle mě je však vhodnější využít kombinaci normování obrazu s technikou *zvýšení kontrastu*. Tím budeme mít opět zaručeno, že všechny hodnoty pixelů v obrazu ve stupních šedi resp. všechny hodnoty jednotlivých barevných složek budou ležet v intervalu $\langle 0, 1 \rangle$ a navíc nejmenší hodnota pixelu v obrazu ve stupních šedi resp. nejmenší hodnota jednotlivé barevné složky bude rovna 0, *vice versa* největší bude rovna 1.

Kýženého výsledku dosáhneme aplikací následujícího vzorečku na každou hodnotu pixelu

v obrazu ve stupních šedi resp. na každou hodnotu jednotlivé složky vektoru RGB :

$$p_i = \frac{p_i - p_{min}}{p_{max} - p_{min}} \quad (27)$$

kde p_{min} je minimální hodnota pixelu obrazu ve stupních šedi resp. každé jednotlivé složky vektoru RGB , vice versa p_{max} je maximální.

3.4 Laplaceovy matice a jejich vlastnosti

V předchozí kapitole jsme naznačili, že diskretní diferenciální operátor Δ na nějaké funkci f bývá často označován jako *Laplaceova matice*. V této kapitole si představíme několik typů těchto specifických matic a vypíchneme jejich nejdůležitější vlastnosti, které budeme dále využívat v teorii *spektrálního shlukování*.

Jistě stojí za zmínku, že mezi první průkopníky spektrální teorie *Laplaceových matic* patří i český vědec M. Fiedler, který v roce 1973 zavedl pojem algebraická souvislost a objevil, že pro dělení grafu na dvě partity je vhodné využít vlastností vlastního vektoru příslušného druhému nejmenšímu vlastnímu číslu.

Za těch více než 40 let naše znalosti o vlastních číslech a vektorech *Laplaceových matic*, nepředpokládejme zatím jejich normalitu, pokročily a začíná se ukazovat, že má smysl zkoumat nejen vlastnosti druhého nejmenšího vlastního vektoru, ale pěkné vlastnosti mají i vlastní vektory příslušné 0 vlastním číslům například, že jsou řešením diskretní *Laplaceovy rovnice*. Dokonce má smysl zabývat se i vlastními vektory příslušných nenulovým vlastním číslům, které jsou analogií k diskretním řešením nehomogenní *Laplaceovy rovnice* tj. *Poissonova rovnice*. Proto můžeme popisovat vlastnosti nenormované *Laplaceovy matice* nejen z pohledu *teorie grafů*, ale pro hlubší pochopení některých zákonitostí je vhodné hledat analogie i v technikách *parciálních diferenciálních rovnic*.

Normovaný tvar *Laplaceovy matice* zase svými vlastnostmi dobře popisuje stochastický proces *náhodné procházky* a můžeme ji chápat jako *Markovovu matici pravděpodobnosti přechodu* náhodného chodce z jedné stavové pozice na ostatní pozice. V informatice bývá náhodná procházka zobrazována pomocí *bezpečného konečného automatu*, ve kterém je ke každému přechodu mezi stavy přiřazena pravděpodobnost s jakou náhodný chodec přejde z aktuálního stavu do následujícího. Ve statistice bývá tento systém stavů a přechodů označován jako *Markovovský řetězec*.

3.4.1 Nenormovaná Laplaceova matice

Nenormovaná Laplaceova matice L je definována jako rozdíl diagonální matice D (25) a matice sousednosti W (24) podobnostního grafu G_s

$$L = D - W. \quad (28)$$

Mezi její nejpodstatnější vlastnosti patří, že je *pozitivně-semidefinitní*, *symetrická* a *singulární*. Nejmenší vlastní číslo nenormované Laplaceovy matice je rovno 0 a jeho násobnost je rovna počtu komponent souvislosti podobnostního grafu G_s [1, str. 4] [3, str. 8].

V aplikaci segmentace obrazu budeme v ideálním případě považovat každou jednotlivou komponentu souvislosti za objekt na obrazové scéně.

Pro vlastní vektory příslušné nulovému vlastnímu číslu *nenormované Laplaceovy matice*, pak platí následujících několik vlastností.

- a) Pro násobnost $k = 1$ nulového vlastního čísla je příslušný vlastní vektor konstantní a odpovídající *indikátorový vektor* f je roven jednotkovému vektoru o velikosti $|V|$.

V některých odborných článcích se můžeme setkat, že *indikátorový vektor* je považován za vlastní vektor příslušný nulovému vlastnímu číslu. Takový případ nastává pouze teoreticky a v praktických aplikacích obdržíme tento vlastní vektor ve tvaru $c \cdot f$, $c \in \mathbb{R}$. Proto chápeme vlastní vektor příslušný nulovému číslu *Laplaceovy matice* za analogii k indikátorovému vektoru f .

- b) Pro násobnost $k \geq 2$ nulového vlastního čísla mají indikátorové vektory f_{A_j} hodnoty prvků $f_{A_j i} = 1$, pokud vrchol v_i patří do komponenty souvislosti A_j , v ostatních případech jsou hodnoty prvků indikátorového vektoru f_{A_j} rovny 0.

Opět teoreticky můžeme předpokládat, že i -tý vlastní vektor příslušný i -tému nulovému vlastnímu číslu je roven i -tému indikátorovému vektoru. V praktických aplikacích tato situace nenastává a výsledný vlastní vektor dostáváme většinou ve tvaru

$$x_i = \sum_{j=1}^k c_j \cdot f_j \text{ kde } c_j \in \mathbb{R}.$$

Důkaz.

Zaveďme si diskretní analogii funkcionálu (11) [3, str. 9]:

$$\int_{\psi \setminus B} |\nabla f(x, y)|^2 dS \sim f^T L f \quad (29)$$

Podle definic L , W , D a f pak můžeme postupovat následovně:

$$f^T L f = f^T D f - f^T W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i=1, j=1}^n f_i f_j w_{ij} = \quad (30)$$

$$= \frac{1}{2} \sum_{i=1}^n d_i f_i - \sum_{i=1, j=1}^n f_i f_j w_{ij} + \frac{1}{2} \sum_{i=1}^n d_i f_i = \quad (31)$$

$$= \frac{1}{2} \left(\sum_{i=1}^n d_i f_i - 2 \sum_{i=1, j=1}^n f_i f_j w_{ij} + \sum_{i=1}^n d_i f_i \right) \stackrel{\text{BÚNO}}{=} \quad (32)$$

$$\stackrel{\text{BÚNO}}{=} \frac{1}{2} \left(\sum_{i=1}^n f_i^2 - 2 \sum_{i=1, j=1}^n f_i f_j + \sum_{j=1}^n f_j^2 \right) \left(\sum_{i=1}^n d_i - 2 \sum_{i=1, j=1}^n w_{ij} + \sum_{j=1}^n d_j \right) \stackrel{\text{BÚNO}}{=} \quad (33)$$

$$\stackrel{\text{BÚNO}}{=} -\frac{1}{2} \sum_{i=1, j=1}^n w_{ij} (f_i - f_j)^2 \quad (34)$$

Dále zaveďme diskretní analogii k minimalizační úloze (12) a předpokládáme, že násobnost nulového vlastního čísla nenormované Laplaceovy matice je rovna 1:

$$0 = -\frac{1}{2} \sum_{i=1, j=1}^n w_{ij} (f_i - f_j)^2 \sim 0 = \sum_{i=1, j=1}^n w_{ij} (f_i - f_j)^2 \quad (35)$$

Aby byla splněna rovnost musí platit:

$$\forall i, j \quad 1 \leq i \leq |V| \wedge 1 \leq j \leq |V| : w_{ij} = 0 \vee (f_i - f_j) = 0 \quad (36)$$

Předpokládáme, že některé vrcholy v_1, v_2, \dots, v_n jsou spojené hranou pak:

$$\sum_{i=1, j=1}^n w_{ij} > 0 \quad (37)$$

takže musí být zaručena druhá podmínka:

$$\sum_{i=1, j=1}^n (f_i - f_j) = 0 \quad (38)$$

to znamená, že všechny hodnoty f_i indexového vektoru f si musí být rovny a podle definice f platí, že f_i se za daných předpokladů musí rovnat 1, tj. f musí být jednotkový vektor velikosti $|V|$ a pak vlastní vektor příslušný nulovému vlastnímu číslu musí být konstantní.

Nyní uvažujme případ, kdy násobnost nulového vlastního čísla $k \geq 2$ tj. podobností graf G_s má ≥ 2 komponenty souvislosti. Bez újmy na obecnosti předpokládáme, že vrcholy podobnostního G_s jsou seřazeny podle příslušnosti k dané komponentě souvislosti. Potom matice sousednosti W má blokovou diagonální formu to samé platí i pro *nenormovanou Laplaceovu matici* L :

$$L = \begin{bmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_{k-1} & \\ & & & & L_k \end{bmatrix} \quad (39)$$

Poznamenejme, že blok L_j je submatice *nenormované Laplaceovy matice* odpovídající j -tému podgrafu G_s , který je j -tou komponentou souvislosti A_j grafu G_s . Podle definice indikátorového vektoru f víme, $f_{A_j i} = 1$ pro ty vrcholy $v_j \in A_i$ ostatní pozice indikátorového vektoru f_{A_i} jsou rovny nule.

Předpokládejme ideální případ, kdy i -tý vlastní vektor příslušný i -tému nulovému vlastnímu číslu se rovná i -tému indikátorovému vektoru komponenty A_i , tj. případ kdy všechny konstanty $c_1, c_2 \dots c_k$ jsou rovny 0, vyjma konstanty c_i příslušné f_i , která je rovna jedné tzn. $x_i = \sum_{j=1}^k c_j \cdot f_j = f_i$.

Dále platí, že každá L_i je opět *nenormovaná Laplaceova matice* souvislého grafu, tj. graf s jednou komponentou souvislosti, proto každá L_i má jedno nulové vlastní číslo a příslušný ideální tvar vlastního vektoru x_i je jednotkový vektor na i -té komponentě souvislosti, jinak je nulový resp. po částech nulový. Proto *nenormovaná Laplaceova matice* má tolik nulových čísel, kolik je komponent souvislosti A_i a příslušné indikátorové vektory f_i jsou analogií ideálního tvaru vlastních vektorů x_i , které jsou příslušné nulovým vlastním číslům *nenormované Laplaceovy matice*. ■

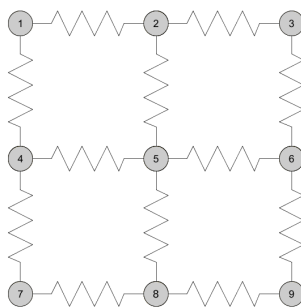
Poznámka Ovšem v praxi je velice obtížné určit, zda má *Laplaceova matice* obrazu násobnost nulového vlastního čísla $k \geq 2$, především v případě úloh, ve kterých segmentujeme

reálné obrazy. U těchto typů úloh je snaha zvolit si relativně malé hodnoty rozptylu σ , tak abychom docílili rozpadu *Laplaceovy matice* na submatice. Avšak vlastní čísla nabývají tak malých hodnot, že je obtížné stanovit, které z nich můžeme považovat za nulové a které už ne. Častým způsobem testování nulovosti vlastních čísel je použití *bartlettova testu homogenity variancí*. Tento přístup však nezaručuje, že jsme opravdu našli nulová vlastní čísla, neboť se jedná o heuristický přístup.

3.4.2 Interpretace vlastních vektorů nenormované Laplaceovy matice

Pokud se díváme na vlastní vektory *nenormované Laplaceovy matice* z pohledu teorie grafů, tak víme, že vlastní vektory příslušné nulovým vlastním číslům nesou informace o komponentách souvislosti *podobnostního grafu* G_s . Má však smysl zkoumat i dříve představenou analogii s řešením *Laplaceovy rovnice*, kde vlastní vektory příslušné nulovým vlastním číslům chápeme jako řešení *diskrétní Laplaceovy rovnice* s nulovými okrajovými podmínkami.

Pro následující úvahy bude vhodnější si představit mřížkový *podobnostní graf* G_s jako soustavu kuliček a pružin, kde vrcholy budeme chápat jako kuličky a hranu jako pružinu mezi dvěma kuličkami. Váhu hrany pak můžeme interpretovat jako tuhost pružiny či lépe jako sílu, kterou na sebe působí dvě kuličky přes pružinu. Potom stupeň vrcholu odpovídá součtu všech sil, které na danou kuličku působí, *vice versa*, ze *zákona akce a reakce*, kterými daná kulička působí na zbývající kuličky. Hodnotu parametru σ budeme chápat jako nějakou vnější okolnost ovlivňující změnu sil působících přes pružiny, vhodným příkladem vnější okolnosti může být například stárí či opotřebení.



Obrázek 6: Podobnostní graf G_s jako soustava kuliček a pružin

Nyní si představme, že přes soustavu kuliček a pružin natáhneme tenkou pružnou membránu a budeme chtít matematicky popsat její model ve stacionárním případě. Nejprve si vyjádříme vertikální výchylky membrány v bodě (x, y) a čase t pomocí funkce $u(x, y, t)$. Dále budeme předpokládat, že na soustavu nepůsobí žádný zdroj sil a soustavu

necháme samovolně kmitat. Po určité době dosáhne soustava rovnovážného stavu, který bude popsán harmonickými funkcemi řešící rovnici

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \Delta u = 0. \quad (40)$$

což, jak víme, je *Laplaceova rovnice*. Pro následující část textu nám stačí vědět, že *Laplaceova rovnice* má nekonečně mnoho řešení. Požadované řešení dostaneme vyšetřením okrajových podmínek. Předpokládejme, že na okrajích membrány nepůsobí žádný zdroj sil, potom okrajové podmínky můžeme považovat za nulové. Pro takto zavedenou úlohu dostáváme, že *Laplaceova rovnice* má jediné řešení a to konstantní funkci. To odpovídá v diskrétní doméně tomu, že *Laplaceova matice* má jediné nulové vlastní číslo s příslušným konstantním vlastním vektorem.

Pokud bychom se snažili pomocí snižování hodnoty rozptylu σ docílit rozpadu *Laplaceovy matice* na několik submatic, potom vlastní vektory příslušné nulovým vlastním číslům odpovídají řešení soustavy *Laplaceových rovnic*. Jak víme z předchozí kapitoly nulovost vlastních čísel je velice obtížné odhadnout a *bartlettův test homogenity variancí* nemůžeme považovat za průkazný.

Předpokládejme, že *Laplaceova matice* má jedno nulové vlastní číslo. Zbývající vlastní čísla *Laplaceovy matice* s příslušnými vlastními vektory řeší vektorovou rovnici

$$L \cdot x = \lambda \cdot x. \quad (41)$$

Pravou stranu rovnice $\lambda \cdot x$ označíme b . Levou stranu rovnice můžeme přepsat pro spojitý případ Δu , kde Δu je spojitě vyjádření diskrétního *Laplaceova operátoru* L . Potom (41) přechází ve spojitě doméně do tvaru

$$\Delta u = b \quad (42)$$

což je *Poissonova rovnice*, která řeší stacionární stav membrány v případě, že na ní působí nějaký časově nezávislý zdroj. Důvod pro vyjádření rovnice (41) ve tvaru spojitě *Poissonovy rovnice* je to, že je potřeba ukázat, že vlastní vektory příslušné nenulovým vlastním číslům *Laplaceovy matice* mohou být diskrétními analogiemi řešení *Poissonovy rovnice*. Proto můžeme při segmentaci obrazu využívat i informace obsažené ve vlastních vektorech příslušné nenulovým vlastním číslům a nedopustíme se až tak závažné chyby.

3.4.3 Normovaná Laplaceova matice, její vlastnosti a interpretace vlastních vektorů

V referenčním textu *A Tutorial on Spectral Clustering* jsou uvedeny dva typy *normovaných Laplaceových matic* L_{sym} a L_{rw} , které svými vlastnostmi popisují stochastický proces náhodné procházky (*random walk*) a můžeme je chápat jako *Markovské matice přechodu*. Matice L_{sym} a L_{rw} jsou svými vlastnostmi navzájem analogické a liší se pouze tím, že první je symetrická a druhá ne. Z praktických důvodů je vhodnější použít symetrickou verzi *normované Laplaceovy matice*, neboť hodnoty vlastních vektorů patří do oboru reálných čísel \mathbb{R} , kdežto v případě nesymetrické L_{rw} mohou nabývat prvky vlastních vektorů hodnot z oboru komplexních čísel \mathbb{C} .

Symetrickou *normovanou Laplaceovu matici* L_{sym} definujeme jako

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (43)$$

a mezi její nejpodstatnější vlastnosti patří, že je *pozitivně-semidefinitní* a *singulární*. Stejně jako *nenormovaná Laplaceova matice* má i L_{sym} nejmenší vlastní číslo rovno 0 a jeho násobnost určuje počet komponent podobnostního grafu G_s s maticí sousednosti $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. V ideálním případě můžeme opět uvažovat každou jednotlivou komponentu souvislosti podobnostního grafu G_s za objekt na obraze.

Pro vlastní vektory příslušné nulovým vlastním číslům *normované Laplaceovy matice* L_{sym} platí teoreticky následující vlastnosti.

- a) Nejprve uvažujme násobnost $k = 1$ nulového vlastního čísla, pak pro vlastní vektor příslušný nulovému vlastnímu číslu *normované Laplaceovy matice* platí:

$$x = D^{\frac{1}{2}} c \quad (44)$$

kde c je konstantní vektor.

- b) Pro teoretickou násobnost $k \geq 2$ nulového vlastního čísla jsou vlastní vektory *normované Laplaceovy matice* příslušné nulovým vlastním číslům ve tvaru:

$$x_i = \sum_{j=1}^k c_j D^{\frac{1}{2}} f_j \quad (45)$$

kde $c_j \in \mathbb{R}$ a f_j je indikátorový vektor j -té komponenty souvislosti A_j podobnostního grafu G_s .

Důkaz.

Důkaz uvedených vlastností *normované Laplaceovy matice* L_{sym} je analogický s důkazem vlastností *nenormované Laplaceovy matice* s tím rozdílem, že se zavádí následující diskrétní analogie funkcionálu (11) [3, str. 9]:

$$\int_{\psi \setminus B} |\nabla f(x, y)|^2 dS \sim f^T L_{sym} f \quad (46)$$

■

Protože *normovaná Laplaceova matice* L_{sym} představuje jistý typ *Markovovy matice přechodu*, můžeme se na vlastní vektory dívat jako na vektory popisující stochastický proces náhodné procházky a hodnoty vlastních vektorů můžeme interpretovat jako pravděpodobnost výskytu chodce na daném pixelu v obrazu.

3.5 Spektrální shlukování jako strategie pro segmentaci obrazu

Cílem této kapitoly je představení metody spektrálního shlukování jako strategie pro segmentaci obrazu. Na rozdíl od klasického přístupu, ve kterém se pro detekci významných oblastí na obrazové scéně využívá pouze hodnot pixelů², je spektrální shlukování založeno na extrakci významných objektů z obrazové scény pomocí informací obsažených ve vlastních vektorech nenormované případně normované Laplaceovy matice obrazu resp. podobnostního grafu obrazu. V teoretickém přístupu k spektrálnímu shlukování využíváme vlastní vektory příslušné nulovým vlastním číslům pro sestavení nových souřadnic dat, v našem případě pixelů, ve vlastním prostoru obrazu a to tak, že vektory uložíme jako sloupce matice $M^{n \times k}$ a za nové souřadnice pixelů p_i považujeme odpovídající řádky y_i této matice. V dalším kroku algoritmu pak pixely v těchto nových souřadnicích shlukujeme. Pro účel této bakalářské práce jsem zvolil jako shlukovací algoritmus heuristický iterativní řešič k-means.

Jak již bylo v textu několikrát zmíněno, ověřit existenci více než jednoho nulového vlastního čísla je obtížný úkol. Položme si tedy otázku, jak správně zvolit počet vlastních vektorů pro sestavení nových souřadnic pixelů. Můžeme postupovat metodou pokus-omyl nebo se můžeme pokusit vhodný počet vlastních vektorů odhadnout. Jako vhodnou techniku pro odhad počtu vlastních vektorů můžeme použít například *Bartlettův test homogeneity variací*, který se v souvislosti se spektrálním shlukováním používá pro detekci nulových vlastních čísel *Laplaceovy matice*. Jak je uvedeno dříve, výsledek *Bartlettova testu* nemůžeme považovat za průkazný, ale může nám dávat vodítko první volby vhodného počtu vlastních vektorů.

Nyní zavedeme dva algoritmy spektrálního shlukování, jejichž vhodnost pro segmentaci obrazu jsem zkoumal v rámci mé bakalářské práce.

²typickým příkladem může být algoritmus k-means

3.5.1 Algoritmus nenormovaného spektrálního shlukování pro segmentaci obrazu

Vstup: digitální obraz, parametr rozptylu σ

1. Normování obrazu: proveď projekci hodnot pixelů obrazu do intervalu $\langle 0, 1 \rangle$.
2. Ze vstupního obrazu sestav podobnostní graf G_s s maticí sousedností W a s ohodnocením hran $w_{ij} = s_{ij}$, kde s_{ij} je míra podobnosti pixelů p_i a p_j určená pomocí Gaussovy podobnosti funkce s parametrem σ .
3. Sestav nenormovanou Laplaceovu matici $L = D - W$.
4. Vypočítej vlastní vektory $x_1, x_2 \dots x_l$ matice L a vhodným způsobem vyber k z nich.
5. Buď $M^{n \times k}$ matice s vlastními vektory $x_1, x_2 \dots x_k$ ve sloupcích.
6. Buď $y_i \in \mathbb{R}^c$ vektor odpovídající i -tému řádku matice M vektorem nových souřadnic pixelů ve vlastním prostoru obrazu.
7. Pomocí algoritmu k -means shlukuj pixely v jejich nových souřadnic y_i .
8. Urči vhodnou barvu výsledných shluků $C_1, C_2 \dots C_k$

Výstup: Shluky $C_1, C_2 \dots C_k$ jako objekty nacházející se na obraze.

3.5.2 Algoritmus normovaného spektrálního shlukování pro segmentaci obrazu

Vstup: digitální obraz, parametr rozptylu σ

1. Normování obrazu: proved' projekci hodnot pixelů obrazu do intervalu $\langle 0, 1 \rangle$.
2. Ze vstupního obrazu sestav podobnostní graf G_s s maticí sousedností W a s ohodnocením hran $w_{ij} = s_{ij}$, kde s_{ij} je míra podobnosti pixelů p_i a p_j určená pomocí Gaussovy podobnosti funkce s parametrem σ .
3. Sestav nenormovanou Laplaceovu matici $L_{sym} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
4. Vypočítej vlastní vektory $x_1, x_2 \dots x_l$ matice L_{sym} a vhodným způsobem vyber k z nich.
5. Buď $M^{n \times k}$ matice s vlastními vektory $x_1, x_2 \dots x_k$ ve sloupcích.
6. Každý řádek matice M nahraď vektorem y_{i_n} , který je jednotkovou normou daného řádku.
7. Buď $y_i \in \mathbb{R}^c$ vektor odpovídající i -tému řádku matice M vektorem nových souřadnic pixelů ve vlastním prostoru obrazu.
8. Pomocí algoritmu k -means shlukuj pixely v jejich nových souřadnic y_i .
9. Urči vhodnou barvu výsledných shluků $C_1, C_2 \dots C_k$

Výstup: Shluky $C_1, C_2 \dots C_k$ jako objekty nacházející se na obraze.

Vhodnou barvu výsledných objektů může vybrat buď segmentátor nebo rozhodnutí může za něj provést počítač. Z praktických důvodů jsem zvolil druhou možnost a segmentační program nastavuje barvu shluku C_i na barvu pixelu, který leží v barycentru tohoto shluku. Je zřejmé, že volba barvy nijak neovlivňuje kvalitu segmentace. Důvod, proč se tedy zabývat volbou vhodného obarvení shluků, je myslím si intuitivní, snažíme se docílit co nejlepší rozpoznatelnosti objektů na grafickém výstupu počítače.

Smyslem mojí práce není podrobné prozkoumání a analyzování uvedených algoritmů, ale praktické poukázání na vhodnost či nevhodnost segmentace obrazu pomocí tohoto přístupu. Další a podrobnější informace ke spektrálnímu shlukování lze nalézt v článku *A Tutorial on Spectral Clustering* od profesorky U. von Luxburk.

3.5.3 Výpočet vlastních čísel a vektorů

Jako numerický řešič pro výpočet vlastních čísel a vektorů *Laplaceových matic* jsem zvolil doporučenou knihovnu ARPACK postavenou na metodě IRAM (*Implicit Restart Arnoldi Method*). V případě výpočtu vlastních čísel a vektorů symetrické matice metoda IRAM konverguje k IRLM (*Implicit Restart Lanczos Method*). Knihovnu ARPACK je možno spustit ve dvou výpočetních módech, *regular* a *shift-invert*.

1. Mód regular

Výpočet je prováděn přímo nad *Laplaceovou maticí*. Výhodou tohoto módu je numerická stabilita. Nevýhodou je pomalá konvergence výpočtu k výslednému řešení.

2. Mód shift-invert

Výpočet je prováděn nad *Laplaceovou maticí* s otočeným spektrem. Výhodou je rychlost konvergence výpočtu k výslednému řešení. Nevýhodou je numerická nestabilita řešení.

Pro výpočet vlastních čísel a vektorů jsem ve své práci používal obě dvě metody a porovnával jsem rychlost výpočtu a numerickou stabilitu řešení. Numerická přesnost vypočítaných vlastních vektorů ovlivňuje kvalitu souřadnic pixelů v novém vlastním prostoru obrazu dimenze k . Výslednou segmentaci obrazu získáme shlukováním pixelů v nových vlastních souřadnicích pomocí algoritmu k -means.

3.5.4 Algoritmus k-means a jeho optimalizace

Algoritmus k -means je heuristický iterativní řešič používaný ve shlukové analýze dat. Pojem k -means byl poprvé použit J. MacQueenem v roce 1967, avšak standardní algoritmus byl představen o 10 let dříve Stuartem Loydem jako technika pulzní modulace. Výsledkem algoritmu je rozdělení n vstupních dat, v našem případě pixelů, do k požadovaných shluků podle vzdálenosti k virtuálním středům, ty jsou vypočítány v každé iteraci jako aritmetický průměr souřadnic bodů uvnitř shluků. Tímto způsobem se rozdělí prostor dat do tzv. *Voronoiho buňek*. V této bakalářské práci jsou využívány pro výslednou segmentaci obrazu 3 modifikace algoritmu k -means: 1. Standardní Lloydův, 2. k -means++ a 3. vlastní modifikace algoritmu pracovně nazvaná jako k -means+2.

Standardní Lloydův Algoritmus Je známý taky jako Voronoiova iterace. Vychází přímo z úvahy o dělení prostoru dat do Voronoiových buňek pomocí vzdálenosti bodů od virtuálních středů. Algoritmus požaduje na vstupu, kromě souřadnic bodů také informaci o počtu požadovaných shluků. Počáteční středy prvotních shluků algoritmus určuje náhodně. Dělí se na dvě části inicializační a aktualizací.

1. **inicializační** - V této části algoritmus postupně prochází prostorem dat a hledá pro jednotlivá data nejbližší virtuální středy.
2. **aktualizační** - Podle nejbližších středů jsou data rozdělena do nových shluků a jsou přepočítány nové středy oblastí.

Iterace těchto dvou částí probíhají dokud není přesun dat mezi shluky dostatečně malý nebo nebyl překročen maximální počet povolených iterací, k-means řeší ve své podstatě minimalizační úlohu

$$\operatorname{argmin} \sum_{i=1}^c \sum_{j=1, x_j \in C_i}^n \|x_j - \mu_i\|^2 \quad (47)$$

kde c je počet požadovaných shluků, x_j jsou body reprezentující data, v našem případě pixely, v prostoru \mathbb{R}^m a μ_i je střed shluku C_i , n je počet vstupních dat.

Vstup: $X = \{x_1, \dots, x_n\}$, počet shluků c , práh ϵ

1. Náhodně vyber počáteční středy shluků.
2. Nastav počáteční vektor příslušnosti U , podle vzdálenosti k počátečním středům.
3. Nastav čítač iterací $b = 0$, nastav $\delta = 0$.
4. Vypočítej středy $\mu_i^{(b)}$

$$\mu_i^{(b)} = \frac{\sum_{j=1 \wedge j \in C_i}^n x_j}{\text{počet bodů v } C_i}.$$
5. Pro všechny body $x_{i=1 \dots n}$ najdi nejbližší střed. Pokud je nejbližší střed stejný jako střed, který přísluší shluku, ve kterém se nachází bod x_i , nedělej nic, jinak zaktualizuj příslušnost u_i a nastav $\delta = \delta + 1$.
6. Pokud $\frac{\delta}{n} < \epsilon$ skonči, jinak nastav $b = b + 1$, $\delta = 0$ a pokračuj krokem 4.

Algoritmus 2: Lloydův algoritmus

Algoritmus k-means++ Oproti standardnímu Lloydovu algoritmu se k-means++ liší v počáteční inicializaci středů a to tak, že první počáteční střed je vybrán náhodně z množiny dat, a každý následující počáteční střed je určen jako nejvzdálenější bod z množiny vstupních dat k posledními vybranému středu.

Algoritmus k-means+2 Ve své podstatě vychází algoritmus *k-means+2* z principu algoritmu *k-means++*, s tím rozdílem, že následující počáteční střed není určen jako nejvzdálenější bod ze vstupní množiny k poslednímu vybranému středu, ale jako nejvzdálenější bod ke všem doposud vybraným počátečním středům.

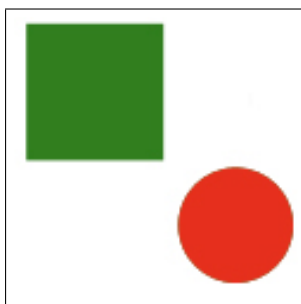
3.6 Dosažené výsledky

V této kapitole představím dosažené výsledky segmentace na různě komplikovaných vstupních obrazech pomocí metod *spektrálního shlukování*. Experimenty jsem prováděl ve dvou barevných úrovních téhož vstupního obrazu. Nejprve jsem segmentoval originální barevný obraz, poté jsem jej převedl do odstínů šedi a porovnal dosažené výsledky.

Při experimentech je věnována pozornost také vlivu změny hodnoty rozptylu σ na dosažený výsledek, stejně jako volbě vhodného počtu vlastních vektorů pro sestavení nových souřadnic pixelů ve vlastním prostoru obrazu a volbě optimální varianty shlukovacího algoritmu *k-means*.

Př. 2 Pokud se testuje libovolná segmentační metoda, zkouší se nejprve její možnosti na vhodně zvoleném uměle vytvořeném obraze, ve kterém jsou na obrazové scéně vyobrazeny jednoduché geometrické obrazce.

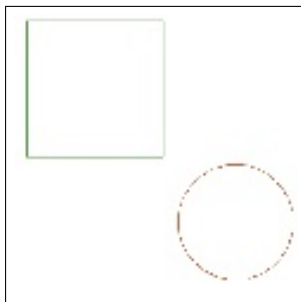
Pro účely testování vhodnosti spektrálního shlukování jako segmentační metody jsem zvolil uměle vytvořený obraz o velikosti 120×120 px, ve kterém je na bílém pozadí zobrazen červený kruh a zelený čtverec.



Segmentace obrazu je subjektivní záležitostí, proto jako součást segmentační úlohy musíme specifikovat, co na obraze hledáme. U takto jednoduchého obrazu není potřeba rozsáhlá diskuze o tom, co budeme považovat za rozumné objekty na obrazové scéně, a intuitivně můžeme zavést segmentační úlohu jako rozdělení obrazu na 3 regiony, které budou reprezentovat zelený čtverec, červený kruh a bílé pozadí.

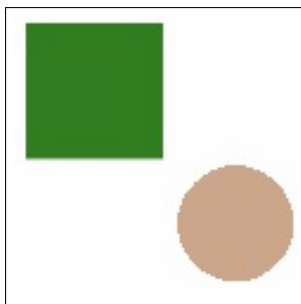
Test metod na obrázku v RGB

1. Nejlepší dosažený výsledek segmentace pomocí metody *nenormovaného spektrálního shlukování*.



Hodnota rozptylu σ	0.05
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	81
Doba výpočtu vlastních vektorů	10min 27s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	25.43s
Počet iterací k-means	2

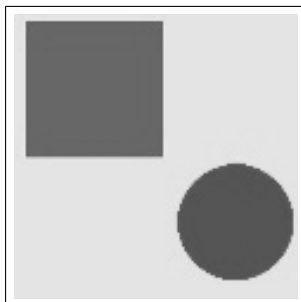
2. Nejlepší dosažený výsledek segmentace pomocí metody *normovaného spektrálního shlukování*.



Hodnota rozptylu σ	0.07
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	4
Doba výpočtu vlastních vektorů	30.24s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	0.18s
Počet iterací k-means	2

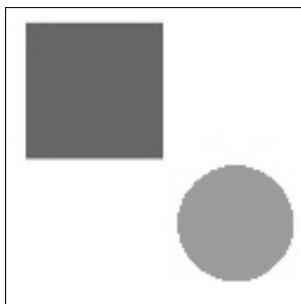
Test metod na obrázku v odstínech šedi

1. Nejlepší dosažený výsledek segmentace pomocí metody *nenormovaného spektrálního shlukování*.



Hodnota rozptylu σ	0.05
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	40
Doba výpočtu vlastních vektorů	1m 44s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	4.55s
Počet iterací k-means	2

2. Nejlepší dosažený výsledek segmentace pomocí metody *normovaného spektrálního shlukování*.



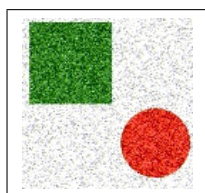
Hodnota rozptylu σ	0.036
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	4
Doba výpočtu vlastních vektorů	31.68s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	0.20s
Počet iterací k-means	2

Zhodnocení testu

Požadovaného a optimálního výsledku jsem dosáhl pomocí normovaného spektrálního shlukování nejen v případě segmentace obrazu v *RGB*, ale i u segmentace obrazu v odstínech šedi. Jak z výsledků testů můžeme vidět nenormovaná varianta spektrálního shlukování je citlivá na nepatrné rozdíly barevných rozdílů mezi pixely a proto se může stát, že budeme potřebovat pro dosažení požadovaného výsledku větší počet vlastní vektorů než v případě normované varianty, případně nemusíme dosáhnout požadovaného výsledku vůbec. Proto v experimentech testující citlivost metody na *Gaussovský šum* se budu zabývat pouze normovanou verzí spektrálního shlukování.

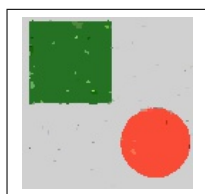
Př. 3 Jelikož normovaná metoda spektrálního shlukování obstála v experimentech na jednoduchém vstupním obrazu, můžeme testovat citlivost této metody na *Gaussovský šum*, který je přidán do uměle vytvořeného obrazu, na kterém jsme prováděli experimenty v Př. 2. Testování se provádělo ve dvou úrovních šumu: středním a vysokém.

1. Střední úroveň šumu *Gaussovského šumu* $\sigma = 0.15$



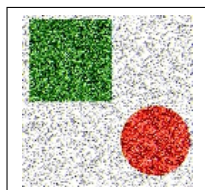
Obrázek 7: Originál vstupu

Nejlepší dosažený výsledek pomocí *normovaného spektrálního shlukování* na vstupu v *RGB*. Metoda byla testována i na vstupním obraze v odstínech šedi, ale nepovedlo se mi najít optimální řešení.



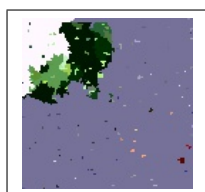
Hodnota rozptylu σ	0.05
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	67
Doba výpočtu vlastních vektorů	1h 1s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	15.74s
Počet iterací k-means	3

2. Vysoká úroveň šumu *Gaussovského šumu* $\sigma = 0.45$



Obrázek 8: Originál vstupu

Pro takto vysokou úroveň *Gaussovského šumu* nebylo možné najít žádné optimální řešení. Nejlepší výsledek, který jsem pomocí metody *normovaného spektrálního shlukování* dostal, je ukázán na následujícím obrázku.



Obrázek 9: Originál vstupu

Hodnota rozptylu σ	0.025
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	126
Doba výpočtu vlastních vektorů	1h 42m
Varianta k-means	k-means+2
Doba běhu algoritmu k-means	59.3s
Počet iterací k-means	5

Př. 4 U reálných obrazů je situace komplikovanější, než v případě uměle vytvořených obrazů. Nejen že je třeba se vypořádat s mělkými přechody mezi pixely tzn. pixely mezi objekty nemají tak velké skoky hodnot, ale barva, popřípadě jas, postupně přechází mezi jednotlivými objekty na obrazové scéně. Navíc je někdy komplikované určit, co budeme považovat za jednotlivé objekty na obraze, proto jsem jako první reálný obraz zvolil



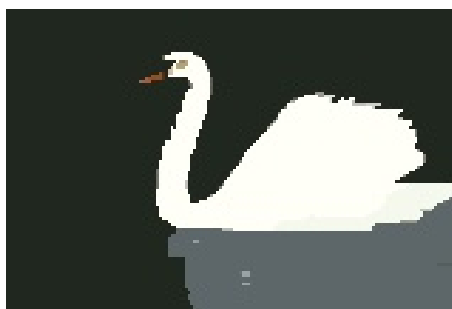
Obrázek 10: 8068.jpg

a jako segmentační úlohu jsem zavedl co nejlepší oddělení labuře od pozadí. Nejprve jsem zkoušel nalézt nejlepší segmentaci na vstupu v originální velikosti 481×321 px. Jelikož pro tak velkou *Laplaceovu matici* obrazu by konvergoval výpočet vlastních čísel a vektorů pomocí knihovny ARPACK v módu *regular* velice pomalu, zvolil jsem výpočetní mód *shift-invert* a opět jsem zvolil normovanou variantu *spektrálního shlukování* a povedlo se mi dosáhnout následujícího nejlepšího výsledku



Hodnota rozptylu σ	0.06
Dimenze Laplaceovy matice	154401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	164
Doba výpočtu vlastních vektorů	5min 46s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	38m 50s
Počet iterací k-means	66

Jelikož pro nižší hodnoty rozptylu σ docházelo k velkým numerickým nepřesnostem při výpočtu vlastního problému *normované Laplaceovy matice*, zmenšil jsem vstupní obraz na velikost 160×107 px a snažil jsem se ověřit, zda je možno nalézt lepší segmentaci pro nižší hodnotu σ . Nejlepší výsledek je opět zobrazen na následujícím obrázku



Hodnota rozptylu σ	0.06
Dimenze Laplaceovy matice	17120
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	34
Doba výpočtu vlastních vektorů	5min 21s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	3.27s
Počet iterací k-means	3

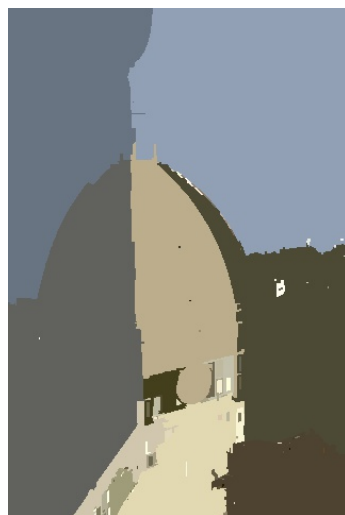
Př. 5 Jako další reálný vstup jsem vybral obraz, na kterém je zachycena scenérie z města.



Obrázek 11: 24004.jpg

Tento obrázek považuji za dostatečně komplikovaný, tak abych prověřil možnosti spektrálního shlukování. Postupoval jsem obdobně jako v předchozím příkladu. Nejprve jsem hledal nejlepší možnou segmentaci vstupu v originální velikosti.

Výpočet vlastních vektorů a čísel *Laplaceovy matice* pro originální velikost vstupu byl opět prováděn v knihovně *ARPACK* v módu shift-invert. Nejlepší výsledek segmentace je zobrazen na následujícím obrázku



Hodnota rozptylu σ	0.02
Dimenze Laplaceovy matice	150401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	106
Doba výpočtu vlastních vektorů	1h 23m
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	10m 22s
Počet iterací k-means	8

Nejlepší dosažená segmentace zmenšeného vstupního obrazu vypadá následovně



Hodnota rozptylu σ	0.02
Dimenze Laplaceovy matice	17120
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	248
Doba výpočtu vlastních vektorů	1h 23m
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	2h 9m
Počet iterací k-means	9

Př. 6 Poslední obraz, který jsem zvolil pro experimenty v této části textu, je následující



Obrázek 12: 27059.jpg

Stejně jako u předchozích testů, ani u tohoto obrazu nemělo smysl ověřovat možnosti *nenormované* verze *spektrálního shlukování*. Nejlepší dosažený výsledek na originálním vstupu je zobrazen na následujícím obrázku a v tabulce jsou opět uvedeny použité vstupní parametry a doby výpočtů jednotlivých procedur.



Hodnota rozptylu σ	0.06
Dimenze Laplaceovy matice	154401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	57
Doba výpočtu vlastních vektorů	27s
Varianta k-means	k-means++
Doba běhu algoritmu k-means na 28 vláknech	1m 59s
Počet iterací k-means	131

Poslední test metody *normovaného spektrálního shlukování* jsem provedl na zmenšeném vstupu o velikosti 160×107 px



Hodnota rozptylu σ	0.03
Dimenze Laplaceovy matice	17120
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	250
Doba výpočtu vlastních vektorů	5h 28m
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 28 vláknech	6m 36s
Počet iterací k-means	9

3.7 Zhodnocení dosažených výsledků

I z této malé množiny testů jsme schopni udělat rozumný závěr. Ukazuje se, že *nenormovaná* verze *spektrálního shlukování* není příliš vhodná pro segmentaci obrazu, neboť je citlivá na malé rozdíly hodnot mezi pixely na jejich lokálním okolí.

Z provedených experimentů vyplývá jako vhodnější metoda pro segmentaci obrazu *normované spektrální shlukování*. Tato metoda dokáže zpracovávat i vstupy zatížené středním *Gaussovským šumem* a při vhodném nastavení vstupních parametrů, vrací optimální řešení segmentace na reálných obrazech. Ovšem i přes klady *normovaného spektrálního shlukování* nedoporučuji tuto metodu pro segmentaci obrazu a to vzhledem k velkým výpočetním nárokům a neintuitivnímu nastavení vstupních parametrů.

4 Difúzní spektrální shlukování jako strategie pro segmentaci obrazu

Difúze je v přírodě přirozený proces, při kterém dochází k vyrovnávání koncentrací látek. To znamená látka s vyšší koncentrací se přesouvá do prostředí s nižší koncentrací dané látky, do té doby než dojde k rovnoměrnému rozložení látky v jednotlivých prostředích.

Difúzní spektrální shlukování je založeno na myšlence vyrovnávání koncentrací barvy či jasů pixelů na ploše obrazu ψ v čase t . Víme, že *Laplaceova* a *Poissonova* rovnice řeší stacionární stav nějaké soustavy, například membránu nataženou na soustavě kuliček a pružin, za předem daných okrajových podmínek a řešení těchto rovnic jsou časově nezávislé. Pokud budeme uvažovat spojitost problému v čase přecházejí *Laplaceova* a *Poissonova* rovnice na difúzní rovnici

$$\frac{\partial u}{\partial t} = \Delta u. \quad (48)$$

Funkci u chápeme jako obrazovou funkci a levá strana rovnice vývoj obrazu v čase, například působením nějakého časově závislého zdroje, pro který můžeme předpokládat, že se v čase ∞ vyčerpá tj.

$$\lim_{t \rightarrow \infty} \frac{\partial u}{\partial t} = 0. \quad (49)$$

Potom difúzní rovnice přechází opět na homogenní *Laplaceovu rovnici*. Pro strojové zpracování je třeba převést spojitě úlohy do diskrétní domény, potom rovnice (48) přechází na tvar

$$\frac{\partial f}{\partial t} = L \cdot f(t), \quad (50)$$

kde L je *Laplaceova matice* a $f(t)$ je *indexový vektor* v čase t . Dá se ukázat, že řešení rovnice (50) je možno nalézt jako

$$f(t) = H(t)f(t_0). \quad (51)$$

$H(t)$ je matice označována jako *diskrétní difúzní operátor* a prvek této matice označuje v naší interpretaci vývoj pružiny, kterou jsou spojené dvě kuličky, v časovém intervalu $[t_0, t]$. A matici $H(t)$ definujeme následovně

$$H(t) = \sum_{k=1}^n e^{-\lambda_k t} u_k u_k^T. \quad (52)$$

kde λ_k a u_k je vlastní číslo *resp.* vlastní vektor Laplaceovy matice L . Potom pro každý vrchol, reprezentující pixel, v podobnostním grafu G_s můžeme definovat nové souřadnice dat ve vlastním prostoru obrazu jako

$$x_i(t) = \left(e^{-\frac{1}{2}\lambda_1 t} u_{i1}, e^{-\lambda_2 t} u_{i2}, \dots, e^{-\frac{1}{2}\lambda_k t} u_{ik} \right) \quad (53)$$

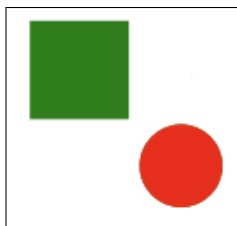
kde parametr t reprezentuje segmentátorem volený čas. V této nové reprezentaci budeme pixely opět shlukovat pomocí algoritmu *k-means*.

Vhodnost metody difúzního spektrálního shlukování pro segmentaci obrazu jsem ověřil na několika následujících příkladech.

4.1 Dosažené výsledky

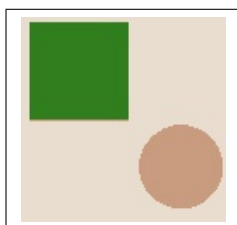
Rozdíl difúzního spektrálního shlukování oproti předcházející metodě je ve své podstatě pouze v přidání časové proměnné t . Proto jsem prováděl následující testy, vždy ve dvou různých časech a pro další parametry metody jsem volil stejné parametry jako u neoptimálnějších výsledků z předchozích experimentů.

Př. 7 Jako první testovaný obrázek pro segmentaci obrazu pomocí *difúzního spektrálního shlukování* jsem zvolil umělý obraz



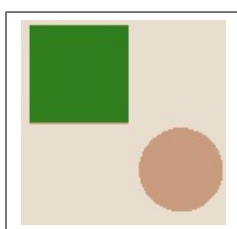
Obrázek 13: scgr.jpg

experiment pro čas $t = 100$



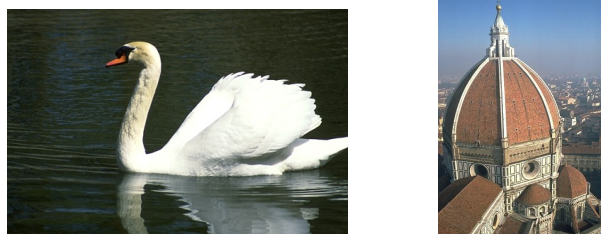
Hodnota rozptylu σ	0.036
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	4
Doba výpočtu vlastních vektorů	31.68s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	0.11s
Počet iterací k-means	2

experiment pro čas $t = 10000$



Hodnota rozptylu σ	0.036
Dimenze Laplaceovy matice	14400
Výpočetní mód knihovny ARPACK	regular
Počet vlastních vektorů	4
Doba výpočtu vlastních vektorů	31.68s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	0.73s
Počet iterací k-means	2

Př. 8 V tomto příkladě si otestujeme metodu *difúzního spektrálního shlukování* na dvou reálných obrázcích.



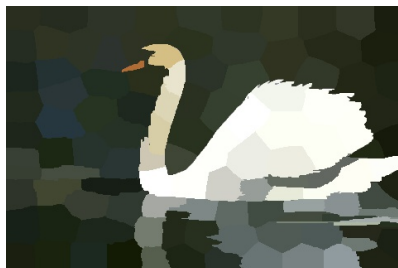
Obrázek 14: Originální obrázky

experiment pro čas $t = 100$



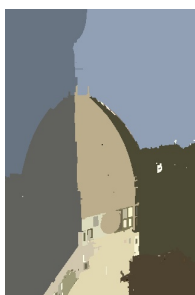
Hodnota rozptylu σ	0.06
Dimenze Laplaceovy matice	154401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	164
Doba výpočtu vlastních vektorů	5m 46s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	44m 50s
Počet iterací k-means	61

experiment pro čas $t = 10000$



Hodnota rozptylu σ	0.06
Dimenze Laplaceovy matice	154401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	164
Doba výpočtu vlastních vektorů	5m 46s
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	41m 55s
Počet iterací k-means	55

experiment pro čas $t = 100$



Hodnota rozptylu σ	0.02
Dimenze Laplaceovy matice	154401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	106
Doba výpočtu vlastních vektorů	1h 23m
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	11m 11s
Počet iterací k-means	13

experiment pro čas $t = 10000$



Hodnota rozptylu σ	0.02
Dimenze Laplaceovy matice	154401
Výpočetní mód knihovny ARPACK	shift-invert
Počet vlastních vektorů	106
Doba výpočtu vlastních vektorů	1h 23m
Varianta k-means	k-means+2
Doba běhu algoritmu k-means na 8 vláknech	10m 1s
Počet iterací k-means	7

4.2 Zhodnocení testů

Jak můžeme vidět z předcházejících testů, při různě zvolených hodnot proměnné t nedochází k výrazným změnám v segmentovaných obrazech oproti spektrálnímu shlukování a to z následujícího důvodu: Hodnota difúzní složky $e^{-\lambda \cdot t} \sim 1$, neboť λ v exponentu je velice malé číslo ~ 0 . Metoda *difúzního spektrálního shlukování* opět není vhodná pro segmentaci obrazu kvůli neintuitivní volbě vstupních parametrů a vysokým výpočetním nárokům.

5 Praktická část

Součástí mojí bakalářské práce bylo i vytvoření softwaru v programovacím jazyce C++, pomocí kterého by bylo možné provádět segmentační experimenty na metodách spektrálního shlukování. Jelikož provádět testy na těchto metodách je časově náročné, rozhodl jsem se vytvořit program, se kterým by bylo možné interaktivně pracovat.

Program jsem vytvořil jako konzolovou aplikaci, která zpracovává příkazy zadávané uživatelem. Pro načítání příkazů ze standardního vstupu byla použita knihovna *GNU Readline*, kontrola a parsování příkazů je prováděno pomocí regulárních výrazů. Pro programátorskou práci s regulárními výrazy mi posloužila knihovna *Boost Regex 1.5.4*. Pro načítání a zpracování obrazových materiálů byla použita knihovna *OpenCV*. Na výpočetní část vlastního problému *Laplaceovy matice* jsem použil knihovny *ARPACK*, *LAPACK* a *BLAS*, o statistickou část se stará software *R* a jeho C++ binding *Rcpp* a *RInside*. Dále bylo třeba v programu vyřešit ukládání dat do lokálního uložiště. Pro ukládání dat jsem zvolil *SQLite databázi*.

```
mari@Methis:~$ ./issc
ISSC - Image Segmentation by Spectral Clustering
version: 1.0 [orlanthi-away], Apr 25 2014
author: Marek Pecha aka Mari
       < marek.pecha.st -at- vsb.cz >
2013-2014 VSB-TUO, Department of Applied Mathematic
>> █
```

6 Závěr

V bakalářské práci jsem ověřil segmentaci obrazu pomocí metod spektrálního shlukování a difúzního spektrálního shlukování. Jelikož segmentace obrazu je velmi subjektivní záležitostí a nedá se objektivně popsat zda výsledky mnou prezentované jsou obecně optimální či nikoliv.

Ve shlukovacím procesu jsem použil následující metody: Standardní Lloydův Algoritmus a k-means++. Při řešení úloh vyvstalo několik problému a to, rychlost výpočtu vlastního problému a určení nulových vlastních čísel *Laplaceových matic*, které popisují podobnostní graf sestavený z obrazu. Rychlost výpočtu jsem vyřešil optimalizováním C++ rozhraní do procedur knihovny *ARPACK*. Problém s určením nulových vlastních čísel jsem vyřešil pomocí Bartlettova testu homogenity variací. V průběhu provádění experimentů se shlukovacím procesem jsem zjistil, že výstupy ze Standardního Lloydova Algoritmu a algoritmu k-means++ jsou z mého pohledu nedostačující. Proto jsem upravil metodu k-means++ následujícím způsobem: navrhl jsem nový způsob výběru počátečních středů shluků. Metoda k-means++ vypočítává následující nejvzdálenější počáteční střed od předcházejícího středu. Moje úprava výpočtu počátečních středů shluků je následující: výpočet následujícího nejvzdálenějšího počátečního středu shluku se určuje jako nejvzdálenější bod ke všem již vybraným středům shluků a tím podle mého subjektivního názoru dostávám optimálnější segmentaci obrazu. Tento algoritmus jsem označil jako k-means+2.

Na základě získaných výsledků můžeme konstatovat, že se metody jeví jako použitelné i pro reálné obrazy avšak složitost nalezení vhodných parametrů pro optimální výsledky, z ní činí pro praktické využití méně vhodnou. Ale i přes výše uvedené se má smysl i nadále vývojem této metody zabývat.

7 Reference

- [1] LUXBURK, Ulrike. *A Tutorial on Spectral Clustering*. 2007.
- [2] KLEINSBER, John. *An Impossibility Theorem for Clustering* . 2002.
- [3] GAURA, Jan. SOJKA, Eduard. *Diffusion-based Image Segmentation Methods*. 2012.
- [4] DOSTÁL Zdeněk. VONDRÁK, Vit. *Lineární algebra*. 2012.
- [5] ARTHUR, David. VASSILVITSKI, Sergei. *k-means++: The Advantages of Careful Seeding*. 2006.
- [6] DRÁBEK, Pavel. HOLUBOVÁ Gabriela *Parciální diferenciální rovnice*. 2011.

A**Příloha**

DVD

Obsah:

1. zdrojové kódy programu
2. elektronická podoba bakalářské práce